

REMARKS

In the Official Action mailed on **18 May 2005** the Examiner reviewed Claims 1-45. Claims 1, 16, and 31 were rejected under 35 U.S.C. §103(a) as being unpatentable over Sexton et al (USPN 6,829,761, hereinafter “Sexton”), further in view of Jardine et al (USPN 6,195,754, hereinafter “Jardine”). Claims 2-5, 8-10, 17-20, 23-25, 32-35, and 38-40 were rejected under 35 U.S.C. §103(a) as being unpatentable over Sexton, in view of Jardine, further in view of Sowizral et al (USPN 6,570,564, hereinafter “Sowizral”) and further in view of Brundridge (USPN 6,279,109, hereinafter “Brundridge”). Claims 6-7, 21-22, and 36-37 were rejected under 35 U.S.C. §103(a) as being unpatentable over Sexton, in view of Jardine, further in view of Sowizral, further in view of Brundridge, and further in view of Danforth (USPN 6,085,034, hereinafter “Danforth”). Claims 11-15, 26-30, and 41-45 were rejected under 35 U.S.C. §103(a) as being unpatentable over Sexton, in view of Jardine, further in view of Sowizral, further in view of Brundridge, and further in view of Bak et al (USPN 6,704,927, hereinafter “Bak”).

Rejections under 35 U.S.C. §103(a)

Claims 1, 16, and 31 were rejected as being unpatentable over Sexton, further in view of Jardine. Applicant respectfully points out that the combined system of Sexton and Jardine teaches using **static initializer code**, which is executable code, to initialize static class variables (see Sexton, col. 8, lines 49-67).

In contrast, the present invention augments the shared runtime data structure with an **initializer field variable**, which is **not** executable code, and uses the value of this variable to determine whether a platform-independent instruction of the program method may trigger an initialization of the class (see paragraphs [0034]-[0043] of the instant application). This is different than using static initializer code to initialize static class variables because it simplifies the

determination of whether a given class is a bootstrap class, and the determination of whether the class initialization was triggered by a class initialization barrier. There is nothing within Sexton or Jardine, either separately or in concert, which suggests augmenting the shared runtime data structure with an initializer field variable, and using the value of this variable to determine whether a platform-independent instruction of the program method may trigger an initialization of the class.

Accordingly, Applicant has amended independent claims 1, 16, and 31 to clarify that the present invention augments the shared runtime data structure with an initializer field variable, and uses the value of this variable to determine whether a platform-independent instruction of the program method may trigger an initialization of the class. These amendments find support in paragraphs [0034]-[0043] of the instant application.

Hence, Applicant respectfully submits that independent claims 1, 16, and 31 as presently amended are in condition for allowance. Applicant also submits that claims 2-15, which depend upon claim 1, claims 17-30, which depend upon claim 16, and claims 32-45, which depend upon claim 31, are for the same reasons in condition for allowance and for reasons of the unique combinations recited in such claims.

CONCLUSION

It is submitted that the present application is presently in form for allowance. Such action is respectfully requested.

Respectfully submitted,

By



Edward J. Grundler
Registration No. 47,615

Date: 1 June 2005

Edward J. Grundler
PARK, VAUGHAN & FLEMING LLP
2820 Fifth Street
Davis, CA 95616-7759
Tel: (530) 759-1663
FAX: (530) 759-1665